UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/767,780	01/28/2004	Shingo Fukui	P/1878-188	5531

2352   7590   05/05/2008
OSTROLENK FABER GERB & SOFFEN
1180 AVENUE OF THE AMERICAS
NEW YORK, NY 100368403

EXAMINER
DARNO, PATRICK A

ART UNIT	PAPER NUMBER
2163	

MAIL DATE	DELIVERY MODE
05/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/767,780	FUKUI, SHINGO
	Examiner	Art Unit	
	PATRICK A. DARNO	2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>16 January 2008</u>.
2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-37</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) <u>1-37</u> is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on <u>28 January 2004</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a) ☒ All   b) ☐ Some *   c) ☐ None of:
      1. ☒ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____.
      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   No new claims have been added. Claims 1, 20, and 37 have been amended. Claims 1-37

are pending in this office action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2.   Claims 1-8, 18, 20-27, 35 and 37 are rejected under 35 U.S.C. 102(b) as being anticipated

by non-patent literature article titled "Improving the Granularity of Access Control for Windows

2000" authored by Michael M. Swift et al. (hereinafter "Swift").

Claim 1:

Swift discloses an information sharing method for holding information owned by at least

one unit user on a storage device in a tree structure provided for each unit user, said tree structure

including a plurality of nodes sequentially arranged from a home root node to a plurality of leaf

nodes, such that said information corresponds to each of said nodes to manage an availability

condition of each of said nodes *(Swift: page 9, section 2.2, lines 4-6 and page 9, Fig. 5 and page 15, section*

3.4, lines 3-5 and page 20, Fig. 11 and page 22, Fig. 13 and page 25, section 4.5, lines 1-3; All citations from the

Swift reference use the page number on the bottom of the article submitted in the Applicant's IDS. The article in its

entirety is 40 pages, the first page being 1 the last being 40.), said method comprising:

a first step in which a computer refers to the availability condition of each of said nodes

on said storage device in response to an availability condition manipulation request for changing

the availability condition of one of said nodes *(Swift: page 20, section 4.2, lines 4-6, and page 20, section 4.2, lines 15-17 and page 22, section 4.3, lines 9-11 and page 22, Fig. 13)*, to determine whether or not said availability condition manipulation request can be executed while satisfying a condition that while the availability condition can be changed at multiple nodes of the tree structure, the number of times of changes in the availability condition is limited to one at maximum along any of paths from said home root node to any one of the plurality of leaf nodes *(Swift: page 11, section 2.4, lines 2-7 and page 16, lines 42-44 and page 17, section 4, lines 15-21 and page 18, lines 22-23 and page 23, lines 1-3 and page 25, section 4.5 and lines 1-10)*;

a second step in which said computer executes the availability condition manipulation request such that said condition is satisfied when the availability condition manipulation request is determined as executable in said first step *(Swift: page 11, section 2.4, lines 2-7 and page 16, lines 42-44 and page 17, section 4, lines 15-21 and page 18, lines 22-23 and page 23, lines 1-3 and page 25, section 4.5 and lines 1-10; It is clear from the references that the access rights (availability conditions) can only be made in one place in the hierarchical structure. Only access manipulations which satisfy this rule are allowed to proceed.)*, and provides a determination that the availability condition manipulation request is not executable when the availability condition manipulation request is determined as not executable in said first step *(Swift: page 21, lines 40-44 and page 25, section 4.5, lines 1-3)*; and

a third step in which said computer refers to said availability condition in response to a tree structure manipulation request for modifying said tree structure, and executes the tree structure manipulation request such that said condition continues to be satisfied *(Swift: page 12, line 20 - page 13, line 3 and page 17, section 4, lines 15-18 and page 18, lines 4-7 and page 20, section 4.2, lines 11-12 and page 22, Fig. 13 and page 25, section 4.5, lines 1-12; It is abundantly clear that the system is restricted to a*

maximum of one availability condition (access control list) change. The reference shows that this condition is maintained whenever changes to the availability condition are requested or the structure is manipulated. Furthermore, it is clear from the references that the system maintains this condition in order to reduce the cost of access time because availability conditions only need to be checked in one place instead of checking availability conditions of each node along a path.).

Claim 2:

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses wherein said first step includes:

when said availability condition manipulation request involves setting an availability condition *(Swift: page 11, section 2.4, lines 2-7 and page 11, section 3, lines 6-8 and page 17, section 4, lines 15-18 and page 25, section 4.5, lines 1-10)*, determining that said availability condition manipulation request is executable when the availability condition of a node under manipulation is the same as that of the home root node, or is a change start point of the availability condition in said tree structure *(Swift: page 11, section 2.4, lines 2-7 and page 16, lines 42-44 and page 17, section 4, lines 15-21 and page 18, lines 22-23 and page 23, lines 1-3 and page 25, section 4.5 and lines 1-10; All of these references reiterate the fact that the availability condition can only be changed a maximum of one time. Since this is the case, at least one of these two conditions MUST be true in order for the predetermined condition of one access change to hold. Specifically, a request to change the availability condition of a node is granted (executable) if the availability condition is the same as the root because (by satisfying the predetermined condition) no change took place yet. And it's granted if it is a change start point because (again by satisfying the predetermined condition) that must be the only change since only one is allowed.)*, and determining that said availability condition manipulation request is not executable when the availability condition of said node under manipulation is

different from that of said home root node, and is not said change start point *(Swift: page 11, section*

2.4, lines 2-7 and page 16, lines 42-44 and page 17, section 4, lines 15-21 and page 18, lines 22-23 and page 23,

lines 1-3 and page 25, section 4.5 and lines 1-10; Again, in order for this the predetermined condition to hold true,

another access change cannot be granted if a node is different from the home root because that means an access

change has already occurred. The same reasoning applies to the change start point. These limitations must occur if

the Swift references inventive feature of one access change on a path of the hierarchy is to hold.).

Claim 3:

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses

wherein said first step includes:

when said availability condition manipulation request involves clearing an availability

condition, determining that said availability condition manipulation request is executable when a

node under manipulation is a change start point of the availability condition in said tree structure,

and determining that said availability condition manipulation request is not executable when said

node under manipulation is not said change start point *(Swift: page 11, section 2.4, lines 2-7 and page 16,*

lines 42-44 and page 17, section 4, lines 15-21 and page 18, lines 22-23 and page 23, lines 1-3 and page 25, section

4.5 and lines 1-10; Again, the predetermined condition of one access change on a path in the hierarchy dictate that

these limitations must occur in the teachings of Swift.).

Claim 4:

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses

wherein said first step includes:

determining that said availability condition manipulation request is not executable when a

node under manipulation intended by said availability condition manipulation request is a home

root node *(Swift: page 21, lines 34-38 and page 21, lines 40-44)*.

<u>Claim 5:</u>

Swift discloses all the elements of claim1, as noted above, and Swift further discloses

wherein said second step includes:

when said availability condition manipulation request involves setting an availability

condition *(Swift: page 11, section 2.4, lines 2-7 and page 11, section 3, lines 6-8 and page 17, section 4, lines 15-*

18 and page 25, section 4.5, lines 1-10), setting the availability condition of a node under manipulation

as requested by said availability condition manipulation request, and setting the same availability

condition to all nodes included in a maximum partial tree in which said node under manipulation

is in a position of a root *(Swift: see at least page 25, section 4.5, lines 1-10; It has been made abundantly clear*

that a maximum of one availability condition occurs in a hierarchy of nodes presented in the swift reference.

Further, it is extremely clear that this access is propagated to all nodes under the node which was changed

(maximum partial tree).).

<u>Claim 6:</u>

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses

wherein said second step includes:

when said availability condition manipulation request involves clearing availability

condition, clearing the availability of a node under manipulation, and setting the same

availability condition as that of said node under manipulation to all nodes included in a

maximum partial tree in which said node under manipulation is in position of a root *(Swift: see at*

least page 25, section 4.5, lines 1-10; Again, it is extremely clear that when an access control change is made (a new

change or deletion/modification of a prior change), the effects of the change are propagated to all nodes under the

node which was changed (maximum partial tree).).

Claim 7:

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses

wherein said third step includes:

when said tree structure manipulation request involves creating a new node, creating said

new node at a requested location *(Swift: page 11, section 3, lines 6-8 and page 12, section 3.1, lines 20-21 and*

page 13, lines 1-3).

Claim 8:

Swift discloses all the elements of claim 7, as noted above, and Swift further discloses

wherein said third step further includes:

setting the same availability condition of a parent node of said new node to said new node

after creating said new node *(Swift: page 18, lines 4-10).*

Claim 18:

Swift discloses all the elements of claim 1, as noted above, and Swift further discloses

wherein each of said nodes in said tree structure is classified into at least one of an unchanged

node having the same availability condition as the home root node; a change start node having an

availability condition different from that of said home root node and different from that of a

parent node; and a change takeover node having an availability condition different from that of

said home root node and the same as that of a parent node *(Swift: page 20, section 4.2, lines 19-21; See*

the INHERITED_ACE flag. This is clearly indication of at least a 'change takeover node'. Based on the

predetermined condition that must be satisfied according to the Swift reference, the Examiner believes the other categories of classification exist in the Swift reference as a result maintaining the availability condition.), said classification being added to information on said availability condition as a change state type of each of said nodes for management, wherein said computer refers to said change state type for examining said availability condition *(Swift: page 20, section 4.2, lines 19-24).*

Claim 20:

Claim 20 is rejected under the same reasons set forth in the rejection of claim 1.

Claim 21:

Claim 21 is rejected under the same reasons set forth in the rejection of claim 2.

Claim 22:

Claim 22 is rejected under the same reasons set forth in the rejection of claim 3.

Claim 23:

Claim 23 is rejected under the same reasons set forth in the rejection of claim 4.

Claim 24:

Claim 24 is rejected under the same reasons set forth in the rejection of claim 5.

Claim 25:

Claim 25 is rejected under the same reasons set forth in the rejection of claim 6.

Claim 26:

Claim 26 is rejected under the same reasons set forth in the rejection of claim 7.

Claim 27:

Claim 27 is rejected under the same reasons set forth in the rejection of claim 8.

Claim 35:

Claim 35 is rejected under the same reasons set forth in the rejection of claim 18.

Claim 37:

Claim 37 is rejected under the same reasons set forth in the rejection of claim 1.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3.   Claims 9-17 and 28-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Swift and further in view of the admitted prior art of U.S. Patent Application Publication

Number 2004/0186845 issued to Shingo Fukui (hereinafter "Fukui APA").

Claim 9:

Swift discloses all the elements of claim 1, as noted above. The Examiner is convinced

that the Swift reference, at the very least, implicitly discloses all the elements of claim 9 as

presented by the Applicant. However, it is noted for the record that the Swift reference does not

appear to explicitly disclose wherein said third step includes:

when said tree structure manipulation request involves duplicating a node group

comprising at least one node, creating a duplicate of said node group at a requested location.

However, Fukui APA discloses:

wherein said third step includes when said tree structure manipulation request involves

duplicating a node group comprising at least one node, creating a duplicate of said node group at

a requested location *(Fukui APA: paragraph [0033] and Fig. 8)*.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Swift with the teachings Fukui APA at the time the invention was made. The skilled artisan would have been motivated to improve the teachings of Swift per the above in order to increase a user's control over the structure of a hierarchical access control system. Furthermore, it is noted that duplicating operations for manipulating a tree structure are admitted by the Applicant to be prior art and, therefore, are clearly well known in the art *(Fukui APA: paragraph [0033] and Fig. 8)*.

Claim 10:

The combination of Swift and Fukui APA discloses all the elements of claim 9, as noted above, and Swift further discloses wherein said third step further includes:

setting the same availability condition set to the parent node of a root node of said node group to said nodes which make up the duplicate of said node group after creating the duplicate of said node group *(Swift: page 18, lines 4-10)*.

Claim 11:

Swift discloses all the elements of claim 1, as noted above. The Examiner is convinced that the Swift reference, at the very least, implicitly discloses all the elements of claim 11 as presented by the Applicant. However, it is noted for the record that the Swift reference does not appear to explicitly disclose wherein said third step includes:

when said tree structure manipulation request involves moving a node group comprising at least one node, moving said node group to a location under a requested destination node.

However, Fukui APA discloses:

when said tree structure manipulation request involves moving a node group comprising

at least one node, moving said node group to a location under a requested destination node *(Fukui*

APA: paragraph [0034] and Fig. 8).

It would have been obvious to one of ordinary skill in the art at the time the invention

was made to modify the teachings of Swift with the teachings Fukui APA at the time the

invention was made. The skilled artisan would have been motivated to improve the teachings of

Swift per the above in order to increase a user's control over the structure of a hierarchical access

control system. Furthermore, it is noted that moving operations for manipulating a tree structure

are admitted by the Applicant to be prior art and, therefore, are clearly well known in the art

(Fukui APA: paragraph [0034] and Fig. 8).

Claim 12:

The combination of Swift and Fukui APA discloses all the elements of claim 11, as noted

above, and Swift further disclose wherein said third step further includes:

performing one processing operation of a plurality of different availability condition

modification processing operations depending on the availability condition of each of said nodes

included in said node group after moving said node group *(Swift: see at least page 18, lines 4-10).*

Claim 13:

The combination of Swift and Fukui APA discloses all the elements of claim 12, as noted

above, and Swift further discloses wherein said plurality of different availability condition

modification processing operations comprises processing for maintaining the availability

condition of each of said nodes included in said node group *(Swift: page 18, lines 4-10 and page 25,*

section 4.5, lines 1-10), processing for setting the same availability condition of said destination

node to each of said nodes *(Swift: page 18, lines 4-10)*, and processing for querying a user whether said processing for maintaining the availability condition or said processing for setting the same availability condition is performed *(Swift: page 21, lines 38-40)*.

Claim 14:

The combination of Swift and Fukui APA discloses all the elements of claim 11, as noted above, and Swift further discloses wherein said third step further includes:

performing one processing operation of a plurality of different availability condition modification processing operations depending on whether the availability condition of said destination node is different from that of the home root node after moving said node group *(Swift: see at least page 18, lines 4-10)*.

Claim 15:

The combination of Swift and Fukui APA discloses all the elements of claim 14, as noted above, and Swift further discloses wherein said plurality of different availability condition modification processing operations comprises processing for maintaining the availability condition of each of said nodes included in said node group *(Swift: page 18, lines 4-10 and page 25, section 4.5, lines 1-10)*, processing for setting the same availability condition of said destination node to each of said nodes *(Swift: page 18, lines 4-10)*, and processing for querying a user whether said processing for maintaining the availability condition or said processing for setting the same availability condition is performed *(Swift: page 21, lines 38-40)*.

Claim 16:

The combination of Swift and Fukui APA discloses all the elements of claim 14, as noted above, and Swift further discloses wherein said third step further includes:

performing one processing operation of a plurality of different availability condition

modification processing operations depending on the availability condition of each of said nodes

included in said node group after moving said node group *(Swift: see at least page 18, lines 4-10).*

Claim 17:

The combination of Swift and Fukui APA discloses all the elements of claim 16, as noted

above, and Swift further discloses wherein said plurality of different availability condition

modification processing operations comprises processing for maintaining the availability

condition of each of said nodes included in said node group *(Swift: page 18, lines 4-10 and page 25,*

section 4.5, lines 1-10), processing for setting the same availability condition of said destination

node to each of said nodes *(Swift: page 18, lines 4-10),* and processing for querying a user whether

said processing for maintaining the availability condition or said processing for setting the same

availability condition is performed *(Swift: page 21, lines 38-40).*

Claim 28:

Claim 28 is rejected under the same reasons set forth in the rejection of claim 9.

Claim 29:

Claim 29 is rejected under the same reasons set forth in the rejection of claim 10.

Claim 30:

Claim 30 is rejected under the same reasons set forth in the rejection of claim 11.

Claim 31:

Claim 31 is rejected under the same reasons set forth in the rejection of claim 12.

Claim 32:

Claim 32 is rejected under the same reasons set forth in the rejection of claim 13.

Claim 33:

Claim 33 is rejected under the same reasons set forth in the rejection of claim 14.

Claim 34:

Claim 34 is rejected under the same reasons set forth in the rejection of claim 15.

4.   Claims 19 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Swift

and further in view of U.S. Patent Application Publication Number 2003/0187854 issued to John

Fairweather (hereinafter "Fairweather").

Claim 19:

Swift discloses all the elements of claim 1, as noted above, but Swift does not appear to

explicitly disclose wherein said tree structure includes a node which is a short-cut to another

node.

However, Fairweather discloses wherein said tree structure includes a node which is a

short-cut to another node *(Fairweather: Fig. 1; Clearly the tree structure in Fig. 1 includes a node 120 which is*

a 'short-cut' to node 130.).

It would have been obvious to one of ordinary skill in the art at the time the invention

was made to modify the teachings of Swift with the teachings of Fairweather noted above. The

skilled artisan would have been motivated to improve the teachings of Swift per the above such

that a short-cut from a given node allows a node to access another node without having to retrace

steps back through the root node. Access time is therefore reduced because fewer nodes have to

be traversed through the tree in order to reach the desired destination node.

Claim 36:

Claim 36 is rejected under the same reasons set forth in the rejection of claim 19.

Response to Arguments

Applicant Argues:

The Examiner believes that the language on page 25, lines 1-3 and 8-10, of Swift is equivalent to the Applicant's limitation that along a path from the home root node to the leaf node the availability condition changes not more than once. Applicant's respectfully disagree. Swift states that the access control changes are made **in one place**, i.e., on a single node of the tree structure.

These multiple changes in access availability are possible because, as Swift repeatedly indicates, "despite presenting data as a hierarchy, the Active Directory internally stores data in a flat database and maintains indexes" over the names and properties of the objects.

Therefore, while the changes are made from a single node of the tree structure, the availability condition along a particular path may change multiple times in the system of Swift.

Examiner Responds:

Examiner is not persuaded. In the rejections of the claims above, the Examiner has cited

more portions of the Swift reference than just page 25, lines 1-3 and 8-10. There appears to be a

multiple embodiments disclosed by Swift reference, and the Applicant may be correct that part

of the reference on page 25 does not teach each and every element of the Applicant's claimed

invention. However, other portions of the reference, and even part of the disclosure of page 25

does in fact appear to anticipate the Applicant's claims.

The Examiner would like to specifically point out a few features of the Swift reference,

and explain why it appears that the Swift reference anticipates the Applicant's claims. The

following references from Swift are now reproduced below. Swift: page 11, section 2.4, lines 4-7

and page 16, lines 42-44 and page 25, lines 1-6.

Note specifically that page 11, section 2.4, lines 4-7 recites, "We also wanted to **allow**

administrators to set access control at a single point in the Active Directory, **and let that policy**

flow to all appropriate objects below that point."

Note specifically that page 16, lines 42-44 recites, "Another potential improvement for evaluating ACLs is to cluster the ACEs in an ACL that grants access to a particular property **to reduce the number of entries that must be inspected**."

Note specifically that page 25, lines 1-6 recites, "Type-specific inheritance and static inheritance allow centralized management **by propagating changes through a hierarchy of objects, so that access control changes are only made in one place**. These features support delegation by allowing an administrator to grant access to a single type of object, or even a single property or single type of object. In addition, **that access is propagated both to existing objects and to new objects when they are created**.

The above passages are cited for a reason. Note that all of most of the passages recited that an access control change is made in one place. More importantly, note that in some embodiments of the Swift reference, <u>this single access control change is propagated down through the access path to all existing and new objects</u>. This means that an access path in a tree structure can only have one access change on any path because once the change occurs the rest of the descendents of that node inherit the access rights.

So for each path, it follows that there can only be a maximum of one change, because when this change occurs, this change is propagated to all descendents that currently exist, and even further, to any descendents that will be created in the future.

Since it appears that each and every element of the Applicant's claimed invention is either disclosed or suggested by the prior art of record, the claims remain rejected under the reasons set forth in the preceding office action. If any of the rejections given above remains

unclear, the Examiner urges the Applicant to schedule an interview to discuss the office action

with the hope of clearing up any ambiguities.

Contact Information

5.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Patrick A. Darno whose telephone number is (571) 272-0788.

The examiner can normally be reached on Monday - Friday, 9:00 am - 5:30 pm.

        If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Don Wong can be reached on (571) 272-1834. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

        Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

                                                        /Patrick A. Darno/
                                                        Examiner
                                                        Art Unit 2163

PAD

/don wong/
Supervisory Patent Examiner, Art Unit 2163